



Bezier Curve Based Medicinal Leaf Classification using Capsule Network

Sandeep Dwarkanath Pande¹, Dr. Manna Sheela Rani Chetty²

¹Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India,
sandeep7887pande@gmail.com

²Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India,
sheelarani_cse@kluniversity.in

ABSTRACT

In this paper Cubic Bezier curve-based leaf classification system using CapsNet is proposed. This system extracts image features in terms of shape characteristics of query leaf image then the CapsNet is trained to classify the input leaf image to a particular class within a desired range of similarity. The proposed system determines nonlinear relationship between leaf image's features for more accurate similarity comparison between query leaf image and existing class images. Among existing approaches to shape feature analysis, statistical approach to extract shape features is adopted here. It works in form of control points of Bezier curves from both given query leaf image and leaf images of available database. These control points are further used to find out the Fourier Descriptors. control points and Fourier descriptors are further used to train the CapsNet for image classification in proposed system. Nowadays, multiple leaf images are generated and stored online hence, classical classification systems are crucial. The proposed system is compared with other classification systems in terms of performance and results. Comparison reveals proposed system performance is state of the art. In many parameters it outperforms other classification systems.

Key words: CNN and Capsule Network, Leaf Classification, Bezier Curves, Control Points.

1. INTRODUCTION

Plants Plays crucial role in our environment. Huge number of plant species are found in the universe. Thousands of them that are found in India having medicinal properties. Near about 80% population of India relies on the therapy of Ayurveda. Nowadays the interest is shifting towards Ayurveda from allopathy and homeopathy. There is a huge knowledge gap in the metropolitan people that they don't know which medicinal plants are useful for which diseases. Some of them knows the plant names and their usefulness but they have never seen those plants. Nowadays as everything is computerised so why should not be a framework that will accept plant leaves and then it will identify and classify the

medicinal plants. There are many reasons that justifies why such a system is crucial like: currently many types of plants are at the risk of extinction. The Flora diversities should be protected, plant databases should be created and their conservation aspects should be recorded, as well to avail their medicinal and ayurvedic properties, plant recognition is required. Most of the plants are used as sources of alternative energy like bio-fuel. Due to these reasons an effective, quick and efficient leaf classification system is always crucial to classify images of user's interest based on query from enormous amount of image database. Leaf classification is also substantial to access visual information and generate knowledge from such information. This promotes for introduction of new leaf classification system which is based on visual contents of the image known as Content Based Leaf Classification (CBLC). CBLC System has a database of leaf images. The system retrieves the images from this database on the basis of input which can be specified by several ways. The input is basically, content of the image i.e. a texture, the shape present in image, color, or combination of these image features. Using resemblance measure the CBLC systems classify the query image to a proper class. Leaf features present in the query image are employed to classify the leaf images.

The constraint of accurate feature extraction and leaf classification is focused here and a new precise approach for modified 2 stage approximation method to extract Control Points (CPs) of a cubic Bezier curve for high variant feature description is proposed. Extraction of accurate CPs from a Bezier curve with an efficient method is also a big challenge. The proposed method uses modified 2 stage algorithm that efficiently computes the CPs of the Bezier curve. The proposed method works better for both concave and convex Bezier curves. In this paper the constraint of optimal feature representation and optimal CapsNet training time is further focused and a new feature vector consisting of Fourier Descriptor (FD) to represent the reduced primary features (i.e. CPs), CPs and 4 extreme points is proposed. The feature vector is further provided as an input for CapsNet that takes reduced training time which in turn results in increased performance as compared to other methods. The proposed method achieves feature selection accuracy and signifies an advantage in low resource overhead in computing shape

feature using Bezier curves and yields accurate results irrespective of position and orientation of the object in image.

The brief review related to the recent researches in Leaf Classification is given in section 2. In section 3, the pre-processing steps required to extract the boundary is presented, also the Bezier curves, and the proposed approach for CP extraction is described. Further it describes representation of shape signatures into FD as an optimal shape feature and the configuration of CapsNet for our approach. The experimental results and setup and the conclusion are discussed in section 4, section 5 respectively.

2. RELATED WORK

Shape, colour, and texture features are employed in most of the preceding researches for plant breed classification by making use of images of leaf. Further, some related studies in this regard are presented in three sections: Content based, Bezier Curve based approaches for image classification, Capsule Neural Networks for image processing.

In [1] the K-NN classifier is used for classification. To classify the shape, and margin-based features such as shape signature in terms of shape descriptor, average positive and negative value, Energy, entropy, magnitude, gradient, curvature petal length and width, sepal length and width, are employed. The probability vectors are then calculated and is used for classification. The method considers vast number of features hence is older and complex approach for leaf classification. An approach using multiscale triangles describing local features of the shape contour points of the leaf; Multiscale Triangular Representation (MTR) is presented in [2]. CNN and machine learning are used in [3] for medicinal leaf classification. The approach uses leaf area, diameter, length, perimeter and width, as feature for CNN and SVM, Naïve Bayes, KNN, Discrimination for classification. The research in [4] uses combination of shape and texture features of leaves for classification. Leaf image detection using shape-based detection is presented in [5]. The approach defines a segmentation technique for detection of leaves using the shape difference and principal component analysis (PCA). The GTCLC [6], combines various features viz. geometric (perimeter ratio, area convexity, circularity), shape (MDM), texture (Sorted Uniform LBP) and color (mean and deviation of hue) features. These features are invariant to scale, illumination and rotation. LDC classifier is used for classification. Feature selection and dimensionality reduction plays a vital role in data analysis tasks. Here multiple features are considered for classification without reduction that affects performance of the approach. In [7] GIST texture features are used for image classification. The method uses PCA for dimensionality reduction. It then uses KNN method for clustering. It employs one-against-one multiclass SVM and Patternet network for pattern recognition and classification. In [8], color histogram in the HSV space, shape feature like majora and minora length, eccentricity and perimeter and

Gray-level co-occurrence matrix (GLCM) as a texture feature. It employs SVM and CNN for pattern recognition and classification. In [9] shape feature Integral Contour Angle (ICA), is used which is translation, rotation and scaling invariant.

Bezier curves [10] are also used as a feature representative curve in image recognition. The method operates on fitting of Bezier curve on the boundary of edge region of an image, and extracting CP representation for this curve. The dominant curvature peaks are used in the CP extraction which is used for feature description. A quaternionic-Bezier curve in three-dimensional space projections for interpolation is proposed in [11]. Here a five-point interpolation or a three point, 2 tangent projections is outlined. The 3D interpolation is used in the image representation where CPs are derived using the Bezier curve representation. This method is having less computation speed as this method involve solution of system of linear equations to compute the CPs. A contour-based leaf classification is presented in [12] that estimate a quadratic curve from centroid, apex and end of petiole. Then it aligns the leaf contour projection on quadratics curve and normalizes mid axis of leaf. The feature selected here is the tangent's direction of leaf boundary. The approach involves many trigonometric and curve approximation computations to reach the feature vector which reduces performance of the system. In [13] various components are combined in hierarchical fashion to form a robust and strong classifier for spectral information. An effective representation method was proposed in [14] which employs convexities and concavities of curvature points. The multi scale convexity concavity defines the feature for different Gaussian kernels and derived as the boundary point of each contour point of the previous level. The approach of curvature coding, results in lower feature descriptors for image retrieval. However, in such coding, features are extracted based on a threshold of the curvature plot, and values with higher magnitude are selected. This approach of feature selection process discards the lower variation information considering as noise. However, in various image samples curvature with variations existing for a lower time period exist. Hence, this assumption of feature selection process minimizes the descriptive features relevancy. In the recognition of CP where images have high varying short period variations, the feature points are large in count. This feature description results in large memory storage and higher search overhead. It also leads to higher probable misclassification in recognition.

CapsNets were introduced recently for image processing tasks [15] that overcome the drawbacks of CNN. Recent works suggest that the features like changing parameters of CapsNet, stacking extra capsule layers gives better results over existing CNNs. In Work [16], CapsNets are studied for better transparency of operations. It is argued that vectorized output of a capsule has likelihood information in regards to features. This enables ease in explain ability of capsule outputs. Next

architecture is Sparse Unsupervised Capsules from [17]. Supervised capsule networks have limitations in depth. Unsupervised capsule networks can be formed deeper layers. Here sparsely connected unsupervised capsules are used for classification problem. System is tested on AFFNIST dataset without training on it. This has given better benchmark results [18]. Fast CapsNet architecture is proposed in [19]. Here performance of CapsNet is improved using limited routing coefficients in each primary capsule. Single pixel in a primary capsule corresponds to single routing path. So, numbers of routing path are drastically reduced. Also, another important contribution of this work is convolutional decoder. This decoder helps in decoding samples with high visual variability in a restricted region [18]. Next, CapsuleGAN architecture [20] is novel step in generative adversarial networks (GAN). Here GAN objective function is also updated [18].

Table 1: Names of Species in dataset

Acer Campestre	Castanea Sativa
Acer Mono	Cornus Chinensis
Acer Opalus	Salix Fragilis
Magnolia_Salicifolia	Salix Intergra
Acer Palmatum	Sorbus Aria
Betula_Austrosinensis	Cytisus Battandieri
Magnolia Heptapeta	Eucalyptus Glaucescens
Quercus_Coccifera	Eucalyptus Neglecta
Alnus Cordata	Eucalyptus Urnigera
Alnus Maximowiczii	Fagus Sylvatica
Alnus Rubra	Ginkgo Biloba
Alnus Sieboldiana	Ilex Aquifolium
Alnus Viridis	Ilex Cornuta
Quercus Afares	Prunus X Shmittii
Zelkova Serrata	Zelkova Serrata

3. METHODOLOGY

The shapes considered in research work are outlines of leaf images which are open or closed curves in a single plane.

3.1 Data

Here, dataset of images of leaves of hundred species is used. Here images without any colours and with pure white background. Coloured images are not part of this dataset. There are total 1600 images for 100 species. List of some species in this dataset is given in table 1. Originally shape, margin and texture feature vectors for each image are given. But, in this work only image data is considered for classification task. The color leaves dataset belongs to James Cope, Paolo Remagnino, Sarah Barman and Thibaut Beghin. The Leaves were taken from the Royal Botanic Gardens, Kew, UK. Originally, sixteen samples per class is the

distribution within dataset. The dataset's purpose is to handle problem of classifying plants based on their leaves. Many of species which are listed here are having medicinal applications. Thus, such classification can help identify medicinal plants efficiently. Data distribution of this dataset is given in table 2.

Table 2: Data distribution of Leaf dataset

	No of Samples	No of Classes	No of Colours
Leaf Dataset	1600	100	2 (Black & White)

Here we can see that each class has very limited number of samples. So, for enrichment of dataset every sample in a class is rotated by 10 degrees between 0 – 360 degrees. Thus, each sample is converted to total 35 samples. Here this rotation of samples is done to also prove the robustness of proposed model for classification. After this enrichment of dataset, all pre-processing steps are carried out.

3.2 Pre-processing

The pre-processing stage outputs the boundary coordinates of shape present in the input image. The steps in pre-processing are shown in Figure 1. The first step is binarization of the shape image that uses simple thresholding to convert the gray level image to binary.

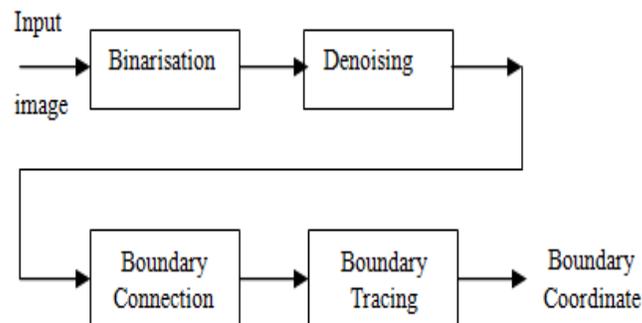


Figure 1: Pre-processing of input image

After thresholding the shapes obtained are sometimes corrupted hence, a denoising process is used to remove noise [21, 22]. Further if the image is blurred or defocussed then various image reconstruction and focussing methods discussed in [23, 24] can be applied to recover the exact image. To get boundary, which is differentiated from rest of the object, following steps are performed on given images.

A. Plotting Histogram

It represents the relative frequency of appearance of numerous gray levels present within an image [25]. The plot obtained between gray levels v/s their frequency is called as gray level Histogram. It is used to obtain approximate level of threshold.

B. Threshold of an image

Threshold plays important role to detect the boundary of an object. Threshold is one of the simplest methods of segmentation. In this, each pixel is assigned to either object part or background part comparing its brightness value to the threshold value.

C. Boundary detection

Boundary detection or contouring is used to extract shapes from given image. In this work, Marching squares algorithm with its latest variations is used for boundary detection [26]. Sometimes, the shape boundary is distorted, therefore, the gaps between boundary points are filled using an 8-connectivity technique [27].

D. Thinning

It converts binary shapes extracted from thresholding to 1-pixel wide lines [21]. After this step, the shapes which are obtained for further processing are irregular shapes and made up of splines. Further, the CPs of the splines are identified using an approximation method.

3.3 Bezier curves and CPs identification

Bezier spline curves have a number of important properties and features that make them highly smooth, useful and convenient for curve and surface drawing. They are also easy to implement [28, 29]. Due to these reasons, Bezier curves are very popular and are widely used in design. Figure 2 shows Bezier curves with 3 and 4 CPs. Let $p_i = (x_i, y_i)$, be the given $m + 1$ control points where i varies from 0 to m . The Bezier curve can be defined using the CPs as: $P(v) = [P_x(v), P_y(v)]$ that outlines the path of a Bezier curve between p_0 and p_n .

$$P(v) = \sum_{i=0}^n p_i BZ_{i,n}(v) \#(1)$$

The $BZ_{i,n}(v)$ are the Bezier blending functions [30] and are given by:

$$BZ_{i,n}(v) = C(n, i)v^i(1 - v)^{n-i}, 0 \leq v \leq 1 \#(2)$$

Where, $C(n, i)$ are the binomial coefficients:

$$C(n, i) = \frac{n!}{i!(n - i)!} \#(3)$$

Cubic Bezier curve, can be described using (1) as:

$$P(v) = p_0B_{0,3}(v) + p_1B_{1,3}(v) + p_2B_{2,3}(v) + p_3B_{3,3}(v) \#(4)$$

Equation (4) can be represented in parametric form using (2) and (3) for computing curve points of a Bezier if CPs are known as:

$$P_x(v) = X_0(1-v)^3 + 3v(1-v)^2X_1 + 3v^2(1-v)X_2 + v^3X_3$$

$$P_y(v) = Y_0(1-v)^3 + 3v(1-v)^2Y_1 + 3v^2(1-v)Y_2 + v^3Y_3, (5)$$

Where (X_0, Y_0) , (X_1, Y_1) , (X_2, Y_2) and (X_3, Y_3) , are the CPs and $0 \leq v \leq 1$.

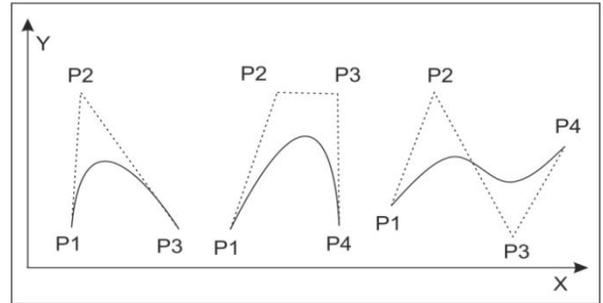


Figure 2: Bezier curves with 3 and 4 CPs

The objective is to find out the exact locations of CPs of a Bezier. This can be achieved by approximating the locations of CPs and computing a new approximated Bezier curve based on approximation. Then determining pointwise RMS error E occurred in approximation where E is the sum of distances between consecutive approximated and actual Bezier curve points. Hence modified two-stage learning algorithm presented below is used to find CPs [30]. Assuming N is the total number points of the Cubic Bezier curve along the shape boundary. Using these curve points, the CPs of the cubic Bezier curve are calculated. Let the 4 CPs to be determined are $p_0, p_1, p_2,$ and p_3 , the i^{th} CP of the original Bezier curve be $P_i(X_i, Y_i)$ and the i^{th} approximate CP be $P'_i(X'_i, Y'_i)$ where, $i = 0, 1, 2, 3, 4$. Steps which are followed are:

A. Modified 1st Stage

It is dedicated to find out the approximate positions of CPs. Here the initial locations of the CPs are determined as:

- a) Determining the 1st and 4th CPs

As per the property of Bezier curves first and last point of sorted curve is the 1st and 4th CP respectively.

- b) Presumption of 2nd and 3rd CPs

Firstly, positions of second and third CPs, are approximated on the tangents to the given curve. CPs lie at vertical distance of H from the base line [29, 30]. Here, H may take one of the $3h/4, h, 4h/3,$ and $1.6h$ values. See Figure 3.

B. Modified 2nd Stage

Here the CPs are recovered. At this stage, finding RMS error E_1 is achieved. E_1 is the summation of the difference between points of given curve $P(v)$ and respective points approximated curve $P'(v)$. In $P'(v)$ CPs, p'_1 and p'_2 are presumed [31]. Bezier curves can be obtained by using (5) [28]. In the equation, the parameter v is between 0 and 1. The value of v is changed by $1/M$ if the given curve is having M points [31]. Suppose the curve $P(v) = [P_x(v), P_y(v)]$, whose CPs to be recovered having $M+1$ points and $P'(v) = [P'_x(v), P'_y(v)]$, be the corresponding points of curve generated by

presumed CPs, where $(v=0, 1/M, 2/M, \dots, 1)$. Thus, the RMS error E_1 is computed by:

$$E_1 \triangleq \sum_{i=0}^M |P(v = M) - q(i)|$$

$$= \sum_{i=0}^M \sqrt{\{P_x(i/M) - P'_x(i/M)\}^2 + \{P_y(i/M) - P'_y(i/M)\}^2} \#(6)$$

Equation (6) is used if the input Bezier curve $P(v)$ is given. The expected results can be calculated by choosing an adequate pair of points of the given and approximated curves [31].

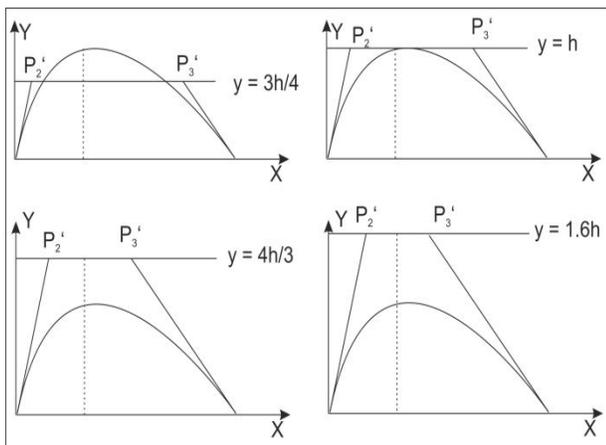


Figure 3: Presumed CPs at different distances from baseline

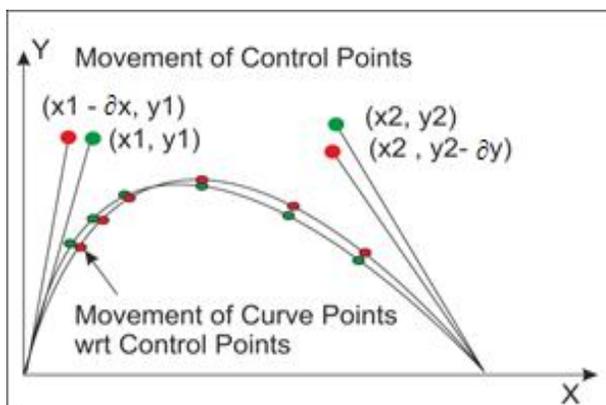


Figure 4: Translation of approximated CPs to minimize RMS error

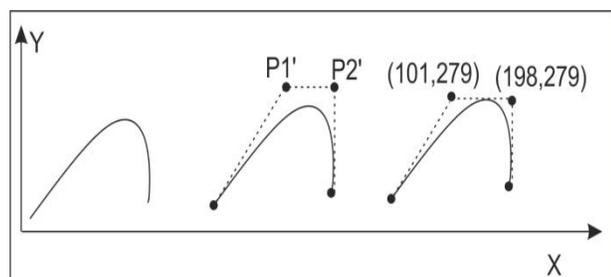


Figure 5: a) Original input curve b) Presumed CPs in modified 1st stage c) Generated CPs in 2nd stage

To recover the CPs, the intermediate CPs are translated along

four directions, one at a time $(x \pm \Delta x, y)$, $(x, y \pm \Delta y)$. If the minimum E_1 is obtained at the position (x, y) , the CPs are not further translated rather the step size Δ for the distance of translation Δ is brought down to half [30, 32]. These iterations are carried until E_1 goes down some allowable (in this approach, 0.000001) value or the learning iteration goes more than the permitted value (in this approach, 90). This translation effect of CPs is shown in Figure 4. The above algorithm gives us CPs for a Bezier (see Figure 5). These CPs are further considered for feature generation like FD [30].

3.4 Shape indexing using Fourier descriptors

Normally, the shape signature is a 1-D function which represents 2-D contours. Here, Central distance function is considered as the shape signature which is calculated for extracted CPs and four extreme points representing dominant points of the curve. It efficiently represents general shapes. The Discrete Fourier Transform (DFT) is applied on the shape signature that generates coefficients identified as optimal feature vector that is FD of the shape. They are frequency domain representation of the shape [30]. Let $r(t)$, $t = 0, 1, \dots, M-1$, be the given normalized shape signature to M points in the sampling stage, the DFT of $r(t)$ is given by:

$$V_N = \frac{1}{M} \sum_{t=0}^{M-1} r(t) e^{\left(\frac{-j2\pi Nt}{M}\right)}, N = 0, 1, 2, \dots, M-1 \#(7)$$

The coefficients V_N , $N = 0, 1 \dots M-1$, are known as FDs of the shape, denoted as FD_N , $N = 0, \dots, M-1$ [30].

3.5 Design of CapsNet

Sequence of multiple convolutional kernels bundled together is known as a capsule. Capsule Networks consists of large number of these capsules. Each capsule operates at local level of features. Global understanding of features is done by communication between various capsules using routing paths. Figure 6 shows architecture of CapsNet as presented in [16]. This section presents how the CapsNet architecture is configured for proposed approach.

A. Building CapsNet and training process

a) Designing the CapsNet

First layer of CapsNet needs to be designed based on input size. In this experiment, input size is kept as 10 by 10. As the input size is small first layer kernel size is set to 3 by 3. Also, reduction in input is not required so stride is kept as 1. Number of outputs from first layer are set to 256. In second layer, 24 primary capsules are used together. Again, kernel or filter size is kept to be 3 by 3. 256 outputs from previous layer act as input to this layer. There can be three stages of routing possible. Higher values of routing require more amounts of processing and results in better pattern selection. So, highest value 3 of routing is used here. Output layer has capsules same to the number of classes in input data. It is a flexible

configuration. If training data is added with a greater number of samples with extra classes then simply by modifying the number of capsules in final layer, new training data can be accommodated in CapsNet training.

b) Training the CapsNet

Training dataset of all images converted in forms of CPs and FDs is used as input to CapsNet. Each image is pre-processed and its CPs along with FDs are extracted. These values in a matrix form act as a single input. So, in training set of all such matrices extracted from all training images is used. Cross Entropy loss is used to optimize CapsNet.

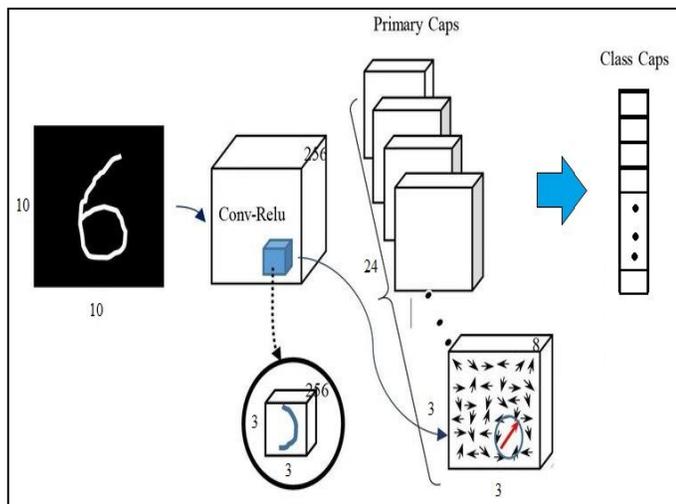


Figure 6: CapsNet Architecture Diagram

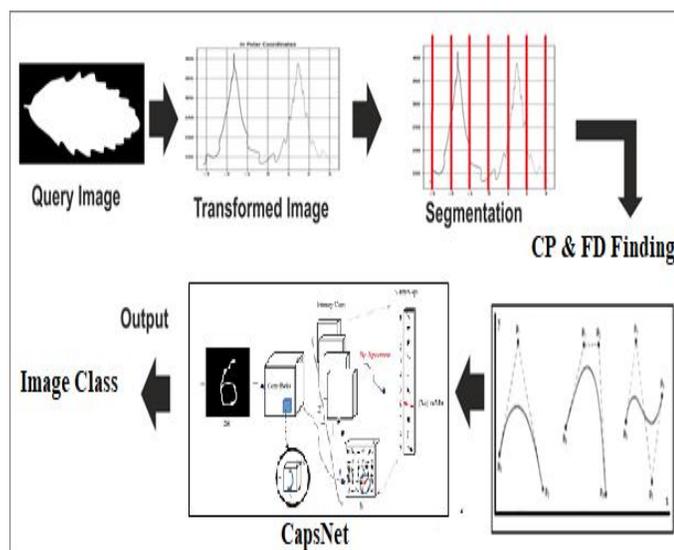


Figure 7: Algorithm Flow

4. EXPERIMENTAL SETUP AND RESULTS

The pre-processing gives set of curve points using gray scale conversion, binarization, denoising, boundary detection Algorithms. This curve is boundary of a shape or object. Now to make the approach position invariant, centre of boundary of every shape is aligned to origin. Then it is converted to linear

curve form. This linear curve is normalized for achieving rotational invariance. The normalized curve is further divided into 10 nearly equivalent segments using segmentation algorithm. Each of these 10 segments can be considered as a separate Bezier curve. Pin pointed approach detects CPs in lesser iterations. The generated 4 CPs in addition to extreme points of the curve are considered in this approach to compute the FD as an optimal feature vector. For each segment 2 such a 20 FDs are calculated. Then these CPs along with FDs are used to train the CapsNet to classify the images. This reduces storage requirement by a large factor. Obviously, such representation leads to efficient comparison at time of matching query images. All steps of proposed method are shown Figure 7. The proposed method is implemented in Python 3.7 with the help of Tensorflow GPU [33] library and Keras software. Simulations are performed on three laptops. First DELL laptop with configuration as, core i5 processor with 2.5 GHz clock speed, 8 GB of RAM, and Ubuntu 18.04 LTS OS. The second DELL laptop with configuration as, Core i7 processor with 2.50 GHz clock speed, 8 GB of RAM and Ubuntu 18.04 LTS OS. The third DELL laptop with configuration as, Core i7 processor with 2.5 GHz clock speed, Nvidia GeForce GTX 860m 4GB GPGPU, 8 GB of RAM and Windows 10 OS. For employing GPGPU to execute proposed method, the system is configured by creating Virtual environment in Anaconda IDE. The performance is evaluated and compared. The reason for configuring CapsNet on GPGPU and CPU for analysing the performance is: it is observed from recent implementations [34] that the Image processing techniques works better on GPUs.

4.1 Evaluation Parameters

The proposed approach's performance is assessed in terms of accuracy. It is the most widely used evaluation parameter for measuring performance of CBLC Systems. It can be defined mathematically as:

$$Accuracy = \frac{\text{Number of correctly classified images}}{\text{Total number of testing images}} * 100\% \quad (8)$$

The proposed approach is compared in the regard of performance with those of various pioneer shape-based image retrieval methods, identified as MTR [2], and Hierarchical approach using multiple descriptors [13]. Multiple iterations are performed on dataset with different samples. Then results are averaged out as mean results. These results are compared with existing techniques and are shown in table 3. Proposed method has high accuracy. The proposed method outperforms in processing time due to the efficient method used for CP generation and also reduced processing time in feature matching due to the use of FDs in CapsNet. Table 4 shows the training time required to execute the proposed approach. It is observed that the proposed approach gives best performance on GPGPU than CPU. The training time required on GPGPU is 10 times lesser than CPU. The comparison of proposed approach is carried out with reserved test samples which are 16.6% of the all leaf images within database.

Table 3: Results from Leaf Dataset

Plant Species Name	No. of Input Images	Images classified Correctly (Proposed)	Images classified Incorrectly (Proposed)	Accuracy (%)		
				MTR [2]	Hierarchical [13]	Proposed
Acer Opalus	35	33	02	93.4	93.8	100
Magnolia_Salicifolia	35	32	03	90.6	90.2	97.5
Acer Palmatum	35	34	01	96	95.1	97.1
Betula_Austrosinensis	35	31	04	86.6	90.6	95.5
Quercus_Afares	35	32	03	89.2	88.2	97.5
Quercus_Coccifera	35	32	02	93.2	90.7	95.5

Table 4: Training Time for CPU and GPGPU

Sr. No.	System Configuration	Training Time (Seconds)
1	Core i5 Processor, 8 GB RAM	1392.8129
2	Core i7 Processor, 8 GB RAM	1151.2021
3	Core i7 Processor, Nvidia GPGPU, 8 GB RAM	140.9176

5. CONCLUSION

This research work has designed and implemented a framework that computes the resemblance of input image with all images in database in regard of the aspect shape characteristics, and returns the class within a sought-after range of resemblance. Here, novel way of using CPs as shape signature of an object is proposed. To detect the CPs new and efficient approach is used. Results show it has acceptable accuracy. Generated CPs are further transformed into optimal Feature vector FD. In this approach from image only FDs and few selected points are selected as input for the CapsNet so it takes less time for training. This approach is position and orientation invariant. Due all these reasons proposed method outperforms others in terms of computation complexity and memory efficiency. Results prove that proposed method achieves feature selection accuracy. This approach yields satisfactory results for leaf databases of limited diversity which merely includes homogeneous patterns. The research work obtains better results and performance compared to other CBLC systems. The proposed approach finds its application in many areas such as; classification and identification of medicinal leaves, Character recognition, digital signature matching, Remote sensing, Handwriting recognition, Ecological learning of leaf, flower, petal etc.

ACKNOWLEDGEMENT

The authors would like to thank K L E F Vaddeswaram, who provided insight and expertise that greatly assisted the research, the infrastructure, tools and opportunity to publish this manuscript.

REFERENCES

1. C. Mallah, J. Cope, and J. Orwell. **Plant Leaf Classification Using Probabilistic Integration of Shape, Texture and Margin Features**, in *Proc. 2013 International Conf. On Signal Processing, Pattern*

- Recognition and Applications, Innsbruck, Austria, 2013, pp. 279-286.
<https://doi.org/10.2316/P.2013.798-098>
2. S. Moune, I. Yahiaoui and A. Verroust. **A Shape-based Approach for Leaf Classification using Multiscale Triangular Representation**, in *Proc. 2013 ACM International Conf. On Multimedia Retrieval*, Dallas, Texas, USA, 2013.
<https://doi.org/10.1145/2461466.2461489>
3. B. Dudi and V Rajesh. **Medicinal Plant Recognition based on CNN and Machine Learning**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 8, No. 4, 2019, pp. 999-1003.
<https://doi.org/10.30534/ijatcse/2019/03842019>
4. J. Chaki, R. Parekh and S. Bhattacharya. **Plant leaf recognition using texture and shape features with neural classifiers**, *Pattern Recognition Letter*, Vol. 58, 2015, pp. 61-68.
<https://doi.org/10.1016/j.patrec.2015.02.010>
5. J. Coussement, K. Steppe, P. Lootens, I. Roldán-Ruiz, and T. Swaef,. **A flexible geometric model for leaf shape descriptions with high accuracy**, *Silva Fennica*, Vol.52, No.2, 2018, pp. 7740:1-7740:14.
6. C. Kalyoncu and Ö. Toygar. **GTCLC: leaf classification method using multiple descriptors**, *IET Computer Vision*, Vol. 10, No.7, pp. 700-708, 2016.
7. F. Kheirkhah and H. Asghari. **Plant leaf classification using GIST texture features**, *IET Computer Vision*, Vol. 13, No. 4, 2019, pp. 369-375.
8. J. Liu, S. Yang, Y. Cheng and Z. Song. **Plant Leaf Classification Based on Deep Learning**, *IEEE International Conf. Chinese Automation Congress*, Xi'an, China, 2018, pp. 3165-3169.
<https://doi.org/10.1109/CAC.2018.8623427>
9. F. Ni and B. Wang. **Integral Contour Angle: An Invariant Shape Descriptor for Classification and Retrieval of Leaf Images**, in *Proc. 25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, 2018, pp. 1223-1227.
10. Y. Chi, C. Chien, and T. Lin. **Leaf Shape Modeling and Analysis Using Geometric Descriptors Derived from**

- Bezier Curves**, *Trans. of the American Society of Agricultural Engineers*, Vol.46, No.1, 2003, pp. 175-185. <https://doi.org/10.13031/2013.12540>
11. S. Zube. **Interpolation method for quaternionic Bézier curves**, in *Proc. Lietuvos Matematikos Rinkiny, Lithuanian Mathematical Society, Ser. A*, Vol. 59, 2018, pp-13-18.
 12. P. Chaisuk, K. Phromsuthirak, and V. Areekul. **Leaf Classification Based on A Quadratic Curved Axis**, in: *Proc. 2017 IEEE International Conf. On Image Processing (ICIP)*, Beijing, China, 2017, pp. 4472-4476.
 13. J. Chaki, R. Parekh, and S. Bhattacharya. **Plant leaf classification using multiple descriptors: A hierarchical approach**, *Journal of King Saud University-Computer and Information Sciences*, 2018. <https://doi.org/10.1016/j.jksuci.2018.01.007>
 14. C. Willcocks, P. Jackson, C. Nelson, and B. Obara. **Extracting 3D Parametric Curves from 2D Images of Helical Objects**, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 39, No.9, 2017, pp. 1757-1769.
 15. S. Sabour, N. Frosst, and G. E. Hinton. **Dynamic routing between capsules**, in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.
 16. A. Shahroudjed, A. Mohammadi, and K. N. Plataniotis. **Improved explainability of capsule networks: Relevance path by agreement**, *arXiv preprint*, 2018, arXiv:1802.10204.
 17. D. Rawlinson, A. Ahmed, and G. Kowadlo. **Sparse unsupervised capsules generalize better**, *arXiv preprint*, 2018, arXiv:1804.06094.
 18. Sandeep Pande and Manna Sheela Rani Chetty. **Analysis of Capsule Network (Capsnet) Architectures and Applications**, *Journal of Advanced Research in Dynamical & Control Systems*, Vol.10, No.10, 2018, pp. 2765-2771.
 19. A. Mobiny and H. Van Nguyen. **Fast capsnet for lung cancer screening**, *arXiv preprint*, 2018 arXiv:1806.07416. https://doi.org/10.1007/978-3-030-00934-2_82
 20. A. Jaiswal, W. Almageed, and P. Natarajan. **CapsuleGAN: Generative adversarial capsule network**, *arXiv preprint*, 2018, arXiv:1802.06167.
 21. K. Gaurav, and P. Bhatia. **A detailed review of feature extraction in image processing systems**, in *Proc. 2010 IEEE 4th International Conf. On Advanced Computing & Communication Technologies (ACCT)*, Rohtak, India, 2014, pp. 5-12.
 22. J. Russ. *The Image Processing Handbook*, 6th ed. Boca Raton, FL: CRC Press, 2016, ISBN-13: 978-1439840450, 2016, ch 6, pp. 105-130.
 23. J. B. Kulkarni and Manna Sheela Rani Chetty. **Depth Analysis of Single View Image Objects based on Object Detection and Focus Measure**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 8, No. 5, 2019, pp. 2608-2612. <https://doi.org/10.30534/ijatcse/2019/112852019>
 24. J. B. Kulkarni and Manna Sheela Rani Chetty. **Comparative Analysis of Various Image Restoration Techniques by measuring Focus Parameter using Discrete Wavelet Transform**, *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 10, No. 10, 2018, pp. 2742-2750.
 25. F. Ciompi, C. Jacobs, E. Scholten, M. Wille, P. Jong, M. Prokop and B. Ginneken. **Bag-of-Frequencies: A Descriptor of Pulmonary Nodules in Computed Tomography Images**, *IEEE Trans. on Medical Imaging*, Vol.34, No.4, 2015, pp. 962-973. <https://doi.org/10.1109/TMI.2014.2371821>
 26. S. Gong, and T. Newman. **Fine feature sensitive marching squares**, *IET Image Processing*, Vol.11, No.9, 2017, pp. 796-802. <https://doi.org/10.1049/iet-ipr.2016.1124>
 27. P. Wong, and Z. Abas. **An Analysis of Human Silhouettes with Normalised Silhouettes Images and Shape Fourier Descriptors**, *International Journal of Human and Technology Interaction (IJHaTI)*, Vol.1, No.1, 2017, pp. 31-35.
 28. F. Gerald. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., ISBN-13: 978-1558607378, 2014, ch. 4, pp. 64-79.
 29. Sandeep Pande, and Manna Sheela Rani Chetty. **Linear Bezier Curve Geometrical Feature Descriptor for Image Recognition**, *Recent Patents on Computer Science*, Vol.12, 2019.
 30. Sandeep Pande, and Manna Sheela Rani Chetty. **Position Invariant Spline Curve Based Image Retrieval Using Control Points**, *International Journal of Intelligent Engineering & Systems*, Vol.12, No.4, 2019, pp. 177-191. <https://doi.org/10.22266/ijies2019.0831.17>
 31. M. Bhuiyan and H. Hama. **Recovering the Control Points of Bezier curves for line image indexing**, *Journal of Electron Imaging*, Vol.11, No.2, 2002, pp. 177-186. <https://doi.org/10.1117/1.1458963>
 32. M. Khan, A. Chowdhury, S. Chowdhury, and K. Deb. **An Efficient Algorithm for Finding the Control Point of 3rd Order Bezier Curve**, in *Proc. IEEE 2010 International Forum on Strategic Technology*, Ulsan, South Korea, 2010, pp:441-445.
 33. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al. **Tensorflow: A system for large-scale machine learning**, in *Proc. 12th USENIX Symposium on Operating Systems Design and Implementation*, Savannah, GA, USA, pp. 265–283, 2016.
 34. J. B. Kulkarni and Manna Sheela Rani Chetty. **Depth Map Generation from Stereoscopic Images using Stereo Matching on GPGPU**, *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 9, No. 18, 2017, pp. 736-747.